Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

Claims 1-16 (*Cancelled*).

17. (*Currently Amended*) A system for describing structure of programming languages, comprising:

(a) first program code written in a ~~high-level~~ first object-oriented programming language, said first program code having first program elements;

(b) second program code written in ~~an extensible,~~ a second object-oriented programming language different from said first object-oriented programming language, said second program code having second program elements,

wherein said second program elements include definitions of objects and descriptions of at least one of inheritance, connections, and encapsulation between said objects, and said objects can be accessed and modified by said first program elements; ~~for describing said first program elements in said first program code written in said high-level programming language~~; and

(c) a programming tool for converting said ~~object-oriented programming language~~ second program code from said second object-oriented programming language to said first object-oriented programming language to produce a converted second program code.

18. (*Currently Amended*) The system of claim 17, wherein copyright text, CCDoc directives, and compiler pragmas are automatically added ~~to the system~~ to said first program code and said converted second program code.

19. (*Currently Amended*) The system of claim 17, wherein input and verification parameters are specified in said ~~extensible and~~ second object-oriented ~~descriptive~~ programming language.

20. (*original*) The system of claim 17, wherein said programming tool is a compiler.

21. (*original*) The system of claim 17, wherein said programming tool is a translator.

22. (*Currently Amended*) A method for describing computer programs by retaining meta-information about program elements, thereby allowing optimization and functionality on multiple hardware and software platforms, comprising the following steps:

(a) creating a first program containing first program elements using a ~~high level~~ first object-oriented programming language;

(b) creating a second ~~corresponding~~ program containing second program elements using ~~an extensible,~~ a second object-oriented programming language different from said first object-oriented programming language, ~~to describe the high~~

~~level source code~~ wherein said second program elements include definitions of objects and descriptions of at least one of inheritance, connections, and encapsulation between said objects, and said objects can be accessed and modified by said first program elements; and

(c) converting ~~the~~ said second ~~corresponding~~ program into a form of ~~the high level~~ said first object-oriented programming language.

23. (*original*) The method of claim 22, wherein the form is machine-executable.

24. (*Currently Amended*) The method of claim 22, wherein the form is ~~high level~~ said first object-oriented programming language.

25. (*Currently Amended*) The method of claim 22, wherein results of said step (a) and said step (b) are placed into one file, ~~and~~ wherein step (c) is preceded by a step of ~~further comprising the steps of:~~

~~(d)~~ copying said second ~~corresponding~~ program from the file into a temporary file, wherein step (c) includes converting said second program of said temporary file into said first object-oriented programming language to produce a converted second program, and wherein step (c) is followed by a step of ~~; and~~

~~(e)~~ combining said converted second ~~corresponding~~ program with said first program ~~the form of the high level source code~~.

26. (*original*) The method of claim 25, wherein the file is a header file.

27. (*original*) The method of claim 26, wherein the header file comprises the following sections:

Definitions;

User Preamble;

User Pre-object;

User Member;

User Postobject; and

User Postamble.